

Requested Patent JP11203160A

Title: TESTING METHOD FOR DATA PROCESSOR ;

Abstracted Patent JP11203160 ;

Publication Date: 1999-07-30 ;

Inventor(s): SHIROTA JOJI; KODAMA YUTAKA; MITSUMATA HIROICHI ;

Applicant(s): HITACHI LTD; HITACHI INFORMATION TECHNOLOGY CO LTD ;

Application Number: JP19980006308 19980116 ;

Priority Number(s): ;

IPC Classification: G06F11/22 ;

Equivalents:

ABSTRACT:

PROBLEM TO BE SOLVED: To provide a testing method for a data processor which avoids the generation of an instruction string that becomes failure due to the same factor and is more efficient with high accuracy. SOLUTION: This method generates a test instruction group through a prescribed test instruction group generation condition with a random number as an input, simulates an instruction of a test instruction group, produces expectation, executes the same instruction on a test object device, gets an execution result, compares the execution result with the expectation, retrieves an instruction that becomes noncoincidence, analyzes the cause and changes a test instruction group generation condition that corresponds to an error cause. Tests for count which designates the operation are carried out and are repeated until they are finished.

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-203160

(43) 公開日 平成11年(1999) 7月30日

(51) Int.Cl.⁶

G 0 6 F 11/22

識別記号

3 1 0

F I

G 0 6 F 11/22

3 1 0 C

審査請求 未請求 請求項の数 1 O L (全 10 頁)

(21) 出願番号 特願平10-6308

(22) 出願日 平成10年(1998) 1月16日

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目 6 番地

(71) 出願人 000153454

株式会社日立インフォメーションテクノロジー

神奈川県秦野市堀山下 1 番地

(72) 発明者 城田 丈治

神奈川県秦野市堀山下 1 番地 株式会社日

立製作所汎用コンピュータ事業部内

(74) 代理人 弁理士 小川 勝男

最終頁に続く

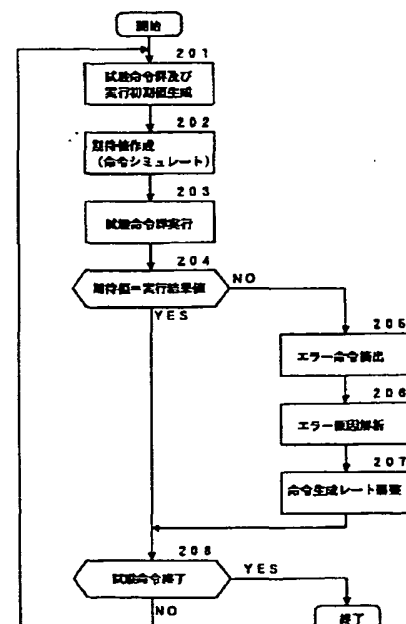
(54) 【発明の名称】 データ処理装置の試験方法

(57) 【要約】

【課題】 同一要因による不良となる命令列の生成を避け、より効率の良い高精度なデータ処理装置の試験方法を提供する。

【解決手段】 乱数データを入力として所定の試験命令群生成条件により試験命令群を生成し、試験命令群の命令をシミュレーションして期待値を作成し、試験対象装置上で同一命令を実行して実行結果を得、実行結果と期待値とを比較して不一致となる命令を検索し、その原因を解析し、エラー原因に対応する試験命令群生成条件を変更する。この操作を指定された回数分の試験を実行し終えるまで繰り返す。

試験装置制御フローチャート (図 2)



【特許請求の範囲】

【請求項1】乱数データを入力として所定の試験命令群生成条件により試験命令群を生成するステップ1と、前記試験命令群の実行結果の期待値をシミュレーションより作成するステップ2と、前記試験命令群を被試験データ処理装置に実行させるステップ3と、前記シミュレーションにより作成した実行結果の期待値と前記データ処理装置における実行結果の値とを比較するステップ4と、比較により不一致が生じた場合に、不一致の要因となった命令及び機能を解析するステップ5と、解析結果により前記試験命令群生成条件を変更するステップ5とからなり、変更された試験命令群生成条件によりステップ1からステップ5の処理を所定回数繰り返すことを特徴とするデータ処理装置の試験方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、データ処理装置の試験方法に関し、特にランダムに試験命令群を生成して被試験データ処理装置の処理機能を試験する方法に関する。

【0002】

【従来の技術】データ処理装置の高性能化を支える技術として、命令をいくつかの処理ステージに分解し、各装置（ユニット）が特定のステージを1マシンサイクル毎に処理するパイプライン処理方式が広く採用されている。この先行制御機能とも呼ばれるパイプライン処理は、複雑かつ大規模な論理によって実現されており、年々先行制御の度合いが非常に深くなっている。その為、先行制御機能の検証は試験命令群の命令数拡大による複雑な機能組み合わせの発生によって、データ処理装置（命令プロセッサ）に高負荷を与えた条件下での試験が重要となっている。

【0003】論理検証を目的とした試験プログラムの中で、乱数データを入力として試験命令群を生成する乱数試験プログラムの従来の技術について説明する。特開平8-166892号には、乱数データを入力として試験命令群を生成する乱数試験プログラムを用いて、パイプライン処理試験を高負荷環境で実行する為の大規模な試験命令を生成し、不一致を発生する命令群を抽出する事で不良となった原因の究明を容易にし、大規模な試験命令の生成を可能とする方法が開示されている。しかし、乱数データを用いて試験を繰り返す行性格上、同一要因による不良を発生する命令群を何度も生成してしまう確率が高く、試験及び不良解析効率が低下してしまうという問題があった。

【0004】

【発明が解決しようとする課題】従来の技術では、乱数データからのみ命令列を生成する為、過去に行った試験を何度も繰り返す結果となる。この為、何度も同一要因に起因する不良を抽出する命令列を生成し同一要因によ

る不良を多発する結果となる。本発明の目的は、上記の課題を解決し、同一要因による不良となる命令列の生成を避け、より効率の良い高精度なデータ処理装置の試験方法を提供するものである。

【0005】

【課題を解決するための手段】上記目的を達成する為、乱数データを入力として所定の試験命令群生成条件により試験命令群を繰り返し生成し、試験命令群を被試験データ処理装置に実行させてデータ処理装置を試験する方法において、シミュレーションにより作成した前記試験命令群の実行結果の期待値と当該データ処理装置における実行結果との比較により不一致が生じた場合、不一致の要因となった命令及び対象機能を見つけ出してエラー原因を解析し、エラー原因に対応する試験命令群生成条件を変更した後、試験を続行する。

【0006】

【発明の実施の形態】以下、本発明の一実施例を図面により詳細に説明する。

【0007】図1は、情報処理装置の試験装置の構成を示す本発明の一実施例のブロック図である。図1において試験装置は、命令生成レートテーブル/機能構築レートテーブル（113）及び乱数値を用いて試験命令詳細情報テーブル（109）及び試験命令群（110）及び実行初期値（111）を生成する試験命令群生成部（101）と、実行初期値（111）を用いて試験命令群（110）をシミュレートして期待値（114）を生成する命令シミュレート部（102）と、実行初期値（111）を初期値とし試験命令群（110）を試験対象の情報処理装置に実行させ実行結果値（115）を得る試験命令群実行部（103）と、期待値（114）と実行結果値（115）を比較して不一致を検出する実行結果比較部（104）と、試験命令群（110）、実行初期値（111）、期待値（114）及び実行結果値（115）からエラーとなる結果を生成する命令を抽出するエラー命令抽出部（105）と、試験命令詳細情報テーブル（109）、試験命令群（110）、実行初期値（111）、命令生成レートテーブル/機能構築レートテーブル（113）、期待値（114）及び実行結果値（115）からエラー原因を解析しエラー解析結果（112）を生成するエラー原因解析部（106）と、エラー解析結果（112）を用いて命令生成レートテーブル/機能構築レートテーブル（113）を更新する命令生成レート調整部（107）と、エラー解析結果（112）、期待値（114）及び実行結果値（115）からエラーメッセージ（116）を作成するエラーメッセージ出力部（108）とより構成される。

【0008】図2は、図1の試験装置全体の処理を示すフローチャートである。命令生成レートテーブル/機能構築レートテーブル及び乱数値を入力として試験命令詳細情報テーブル、試験命令群及び実行初期値を生成し

(201)、試験命令群の命令をシミュレートし期待値を作成する(202)。その後試験対象装置上で同一命令を実行し実行結果値を得(203)、実行結果値と期待値を比較チェックし(204)不一致となる実行結果値を生成する命令を検索する(205)。(205)で検索した命令の結果が不一致となった原因を解析してエラー解析結果を生成する(206)。命令調整レート調整部(207)はエラー原因解析(206)で得られた解析結果を元に、命令生成レートテーブル/機能構築レートテーブルを同一要因不良が出なくなるように更新する。該記操作を指定された回数分の試験を実行し終えるまで繰り返す(208)。

【0009】図3は、本発明の実施例で使用される各テーブルのフォーマットを示す図である。試験命令詳細情報テーブル(301)には、生成された各命令について、命令種別、命令タイプ、フォーマット、使用したレジスタ番号、生成された命令がメモリアクセス命令ならばアクセスしたアドレス等の詳細な情報が書込まれる。機能構築レートテーブル(302)には、生成する命令列が実現する機能の出現頻度が設定されている。命令生成レートテーブルには、生成する命令列が実現しようとする各機能に対応した命令タイプ毎の出現頻度が書込まれたものと(303)、命令タイプ内の命令種別毎の出現頻度とコンフリクト幅とその頻度が書込まれたもの(304)がある。

【0010】図4は、命令生成レート調整部の処理を示すデータフロー図である。FRコンフリクト機能のテストが目的となっていて、フローティング演算命令でパイプライン動作不良となっていた場合(401)、フローティング演算レジスタコンフリクトの試験発生頻度を下げよう機能構築レートを更新し(402)、エラーとなった命令のエラー対象コンフリクト幅の発生頻度を下げよう命令生成レートを更新する(403)。この操作により、エラー原因となる命令パターンの生成を抑止することができる。

【0011】図5は、図2の試験命令群及び実行初期値生成(201)の処理中の試験命令群生成方法を示すデータフローである。機能構築レートテーブルの頻度より生成する命令列が実現する機能を決定し(501)、命令生成レートテーブルでは、選択された機能を実現する為の命令タイプ毎命令出現頻度表を選択し、その頻度より出現させる命令タイプを選択する(502)。命令タイプ毎の命令生成レートテーブル内の命令種別毎の頻度によって生成命令を決定し(503)コンフリクト幅の頻度によって試験命令詳細情報テーブルを決定しターゲットとして使われたリソースデータを元にオペランドを決定する(504)。この時、コンフリクト幅とはあるリソースを更新してから使用するまでに実行される命令数を言う。

【0012】図6は、図2のエラー命令抽出処理(20

5)のフローチャートを示す図である。命令数カウンタ値Nの初期値を1とする(601)。対象テスト命令列の先頭から命令カウンタ値分の命令列を選択し(602)、当該命令列をシミュレート実行して期待値を生成する(603)。その後対象装置上で同一命令を実行し(604)、シミュレートで得た期待値と実行結果を比較チェックする(605)。結果が正しい場合には、命令数カウンタを+1(606)し、同様の処理を繰り返す。結果が不正であった場合、実行した命令列中の最後の命令が障害要因を持つエラー命令であると判断し、処理を終了する。

【0013】上記処理により、対象テスト命令列実行後の被テスト対象装置上の情報(メモリ・レジスタ)では障害部位が特定できないような複雑な動作シーケンスを有するテスト命令列に於いて、障害要因を有するテスト命令を特定することが可能となる。

【0014】図7は、図2のエラー原因解析(206)の処理を示すフローチャートである。TLB機能をoffとし、命令群を実行して実行結果値を生成し(701)、生成した実行結果値と期待値とを比較する(702)。結果が等しければTLB不良によるエラーであることを報告する(715)。結果が等しくない場合は命令コードやオペランドデータをIn-cacheで実行して実行結果値を生成し(703)、生成した実行結果値と期待値とを比較する(704)。結果が等しければメモリアクセス不良によるエラーであることを報告する(714)。結果が等しくない場合はエラー命令の前にパイプライン調整用命令(SYNC/NOP)を挿入して実行結果値を生成し(705)、生成した実行結果値と期待値とを比較する(706)。結果が等しければパイプライン不良によるエラーであることを報告する(713)。結果が等しくない場合は、エラー対象命令が使用しているレジスタ名を変更して実行結果を生成し(707)、生成した実行結果値と期待値とを比較する(708)。結果が等しくない場合はデータ依存タイプのエラーであることを報告する(709)。結果が等しい場合は、エラー命令単体でのデータバリエーション試験を行い(710)、NGが発見されればデータ依存タイプのエラーであることを報告する(709)。NGがない場合は物理障害タイプのエラーであることを報告する(712)。最後に、命令種別、命令タイプ、機能構築データ等をエラー解析結果に書き込む(716)。

【0015】対象装置がデグラモードをサポートしている場合に於ては、同様の障害解析を対象装置の機能で実現可能である。

【0016】

【発明の効果】本発明により、同一要因による不良の発生を回避することによって不良原因解析の工数を低減するとともに、検証精度を向上させることが可能となり、短期間で高精度な論理検証が可能となる。

【図面の簡単な説明】

【図1】本発明の一実施例の試験方法によりデータ処理装置の試験を実施する試験システムの構成を示すブロック図である。

【図2】本試験システム全体の処理を示すフローチャートである。

【図3】本試験システムが使用するデータテーブルフォーマット例である。

【図4】命令生成レート調整部の内部処理を示すブロック図である。

【図5】試験命令生成部の内部処理を示すブロック図である。

【図6】エラー命令抽出部の内部処理を示すフローチャートである。

【図7】エラー原因解析部概略の内部処理を示すフローチャートである。

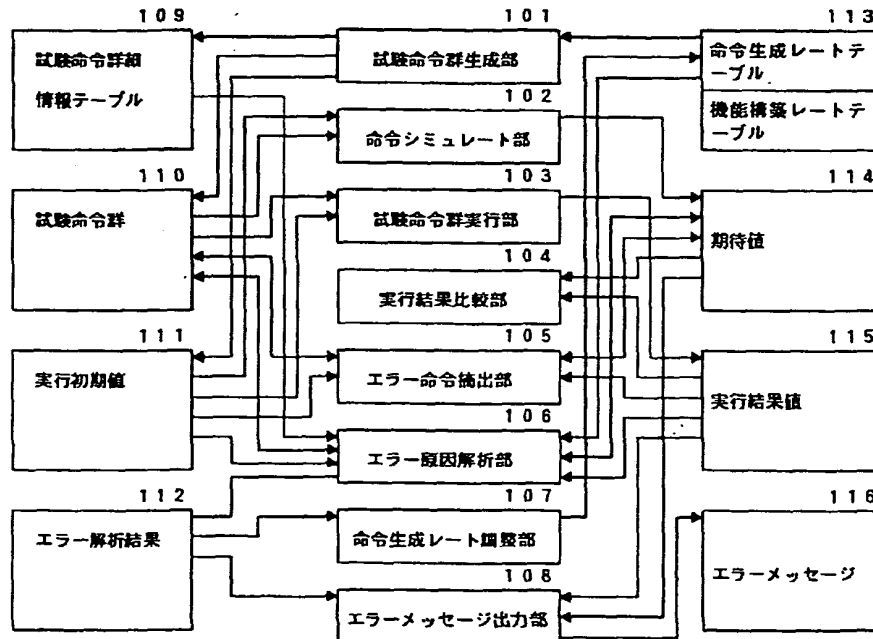
【符号の説明】

- 101 試験命令生成部
- 102 命令シミュレート部
- 103 試験命令群実行部
- 104 実行結果比較部
- 105 エラー命令抽出部
- 106 エラー原因解析部
- 107 命令生成レート調整部
- 108 エラーメッセージ出力部
- 109 試験命令群詳細情報テーブル
- 110 試験命令群
- 111 実行初期値
- 112 エラー解析結果
- 113 命令生成レートテーブル/機能構築レートテーブル
- 114 期待値
- 115 実行結果値
- 116 エラーメッセージ

【図1】

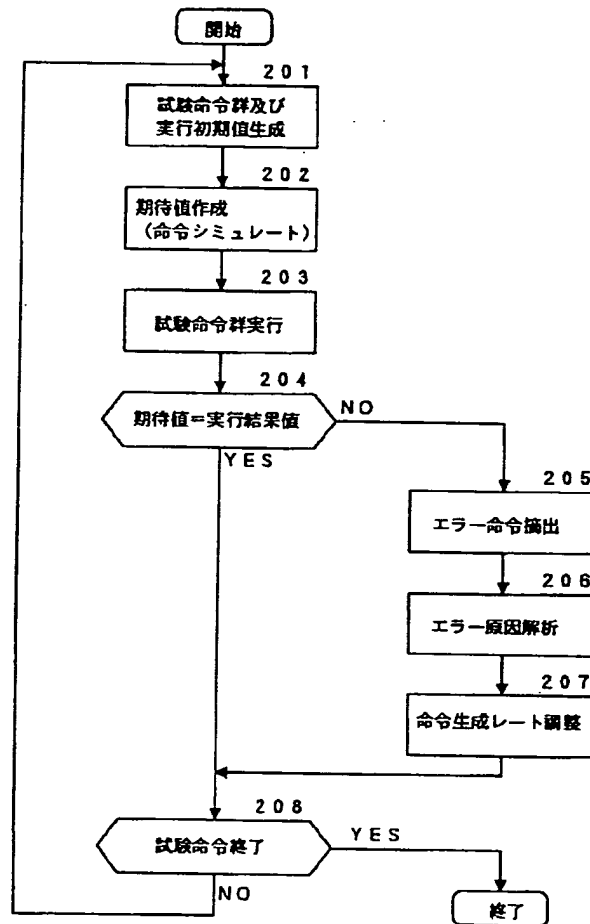
機能構築レート

試験装置ブロック図 (図1)



【図2】

試験装置概略フローチャート (図2)



【図3】

使用テーブルフォーマット例 (図3)

試験命令詳細情報テーブル

301

種別	Add
タイプ	RR
フォーマット	X
使用リソース	St
アドレス	0x100
機能	GR_CNF
...	...

機能構築レートテーブル

302

機能	頻度[%]
GR コンフリクト	20
FR コンフリクト	20
CR コンフリクト	10
PR コンフリクト	5
...	...

命令生成レートテーブル

303

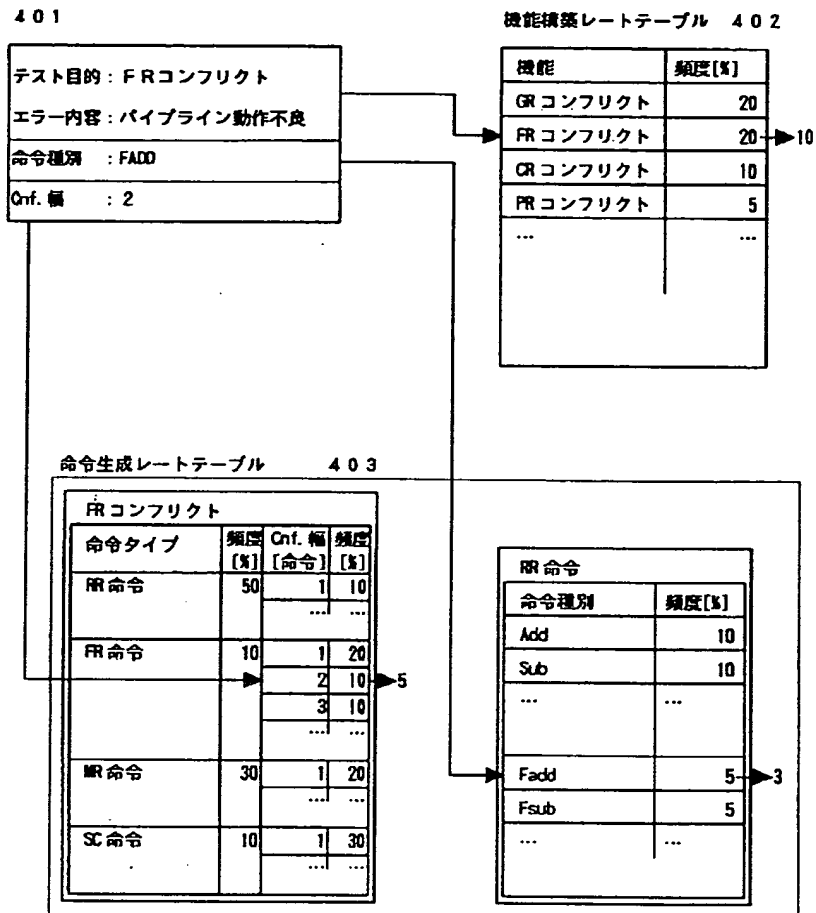
GR コンフリクト			
命令タイプ	頻度 [%]	Cnf. 幅 [命令]	頻度 [%]
RR 命令	50	1	10
		2	10
		3	10
	
FR 命令	10	2	20
	
MR 命令	30	1	20
	
SC 命令	10	1	30
	

304

RR 命令	
命令種別	頻度 [%]
Add	10
Sub	10
Mul	5
Div	5
...	...

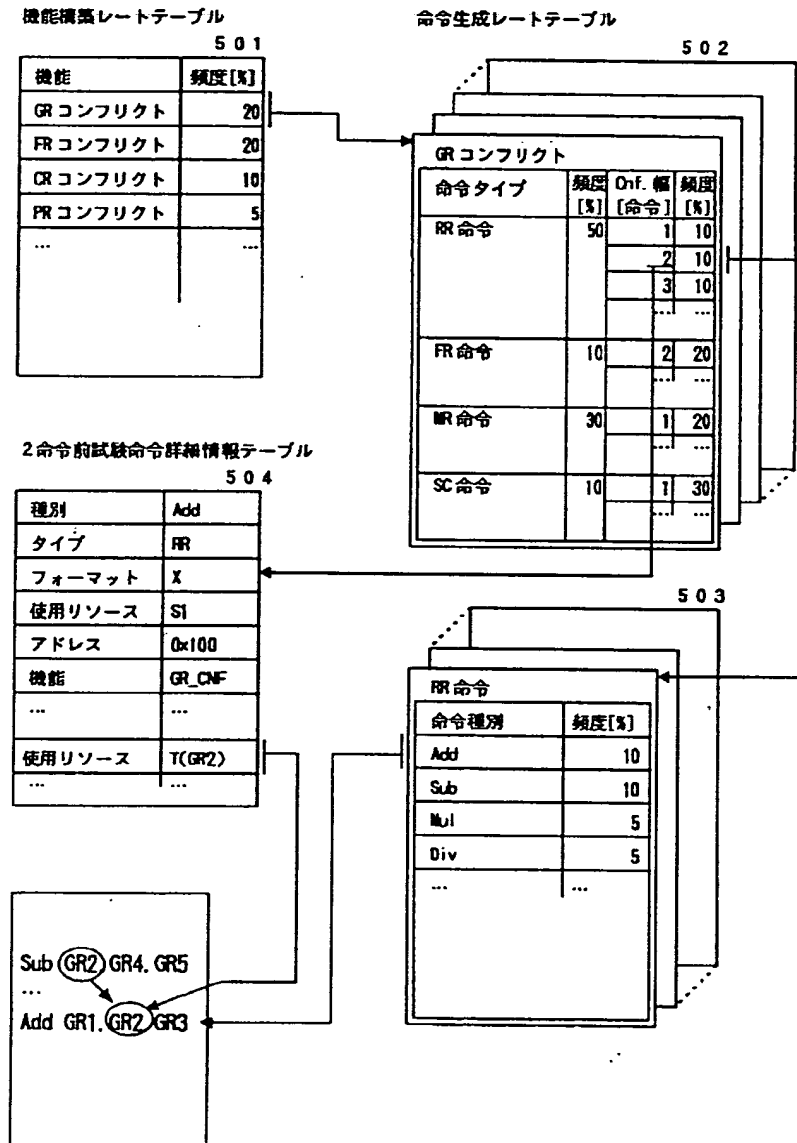
【図4】

命令生成レート調整部の処理概要 (図4)



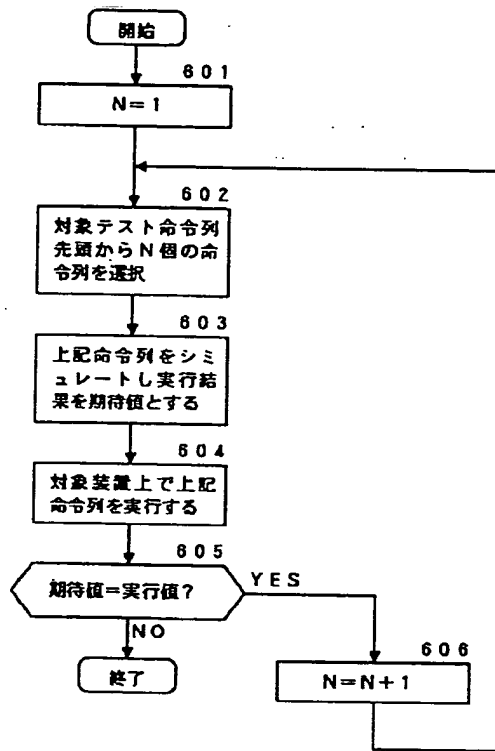
【図5】

命令生成概略ブロック図 (図5)



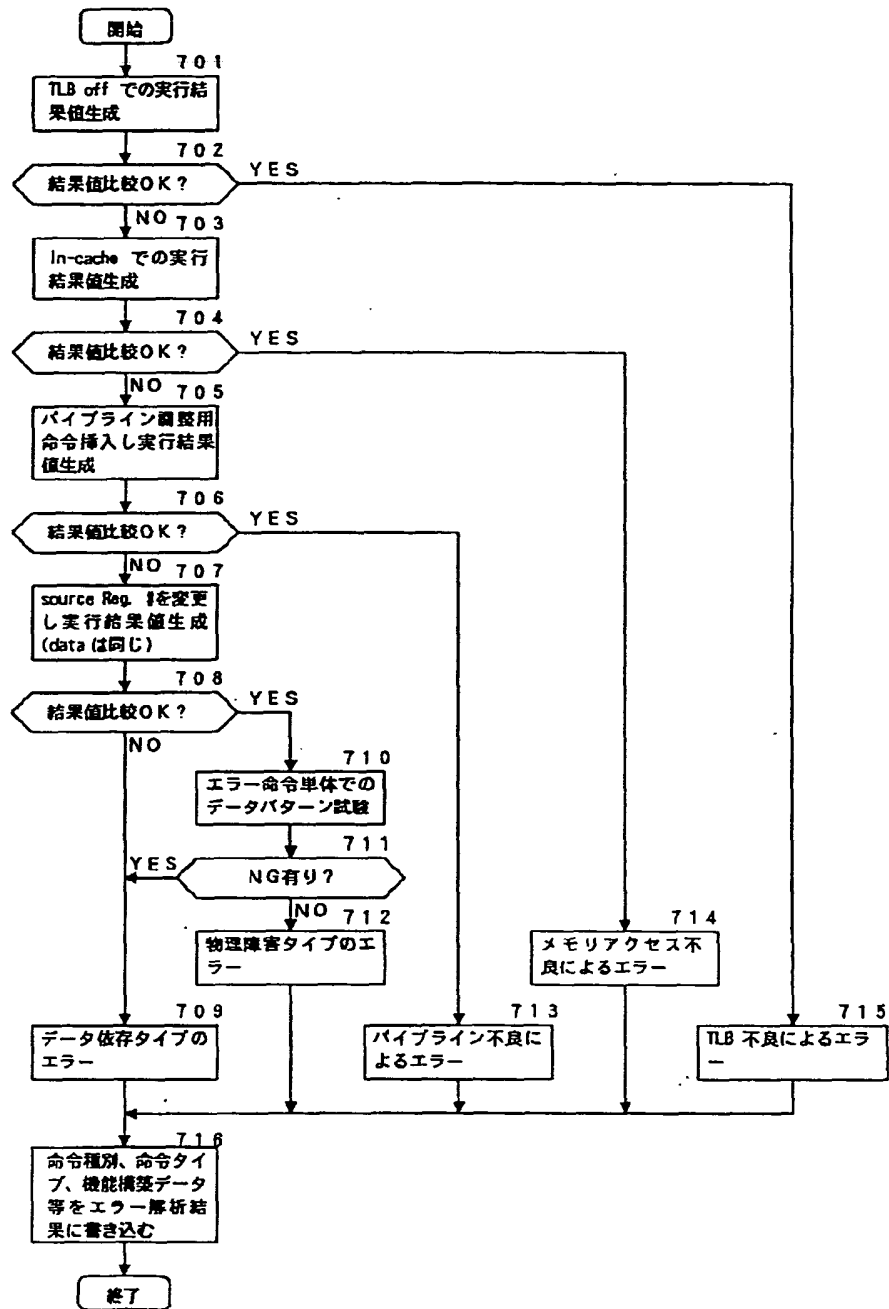
【図6】

エラー命令抽出部概略フローチャート (図6)



【図7】

エラー原因解析部概略フローチャート (図7)



フロントページの続き

(72)発明者 児玉 豊
 神奈川県秦野市堀山下1番地 株式会社日
 立インフォメーションテクノロジー内

(72)発明者 三俣 博一
 神奈川県秦野市堀山下1番地 株式会社日
 立インフォメーションテクノロジー内